Phân loại thư rác sử dụng thuật toán SVM

Nhóm 3

B22DCKH024 - Vũ Công Tuấn Dương
B22DCCN768 - Nguyễn Sơn Tùng
B22DCCN479 - Nguyễn Đức Lâm
B22DCCN347 - Trần Đức Hoàng
B22DCCN348 - Trần Huy Hoàng

Ngày 14 tháng 4 năm 2025

Bộ dữ liệu

Dữ liệu được lấy trên Kaggle

- Combined Spam Email CSV of 2007 TREC Public Spam Corpus and Enron-Spam Dataset
- 83446 bản ghi email bằng tiếng Anh được phân loại thành 2 nhãn là Spam và Non-spam

Mục tiêu

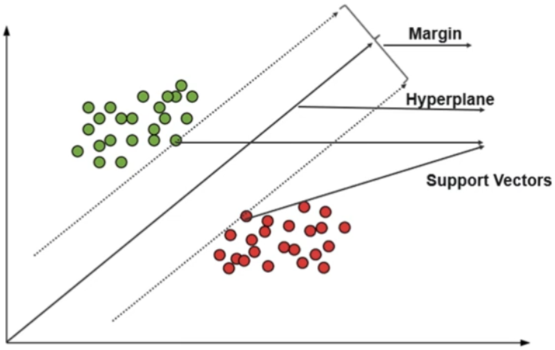
- Xây dựng được mô hình Linear SVM Hard Margin cơ bản để phân loại
- Đánh giá mô hình dựa trên các thang đo như độ chính xác và F1 score
- Demo được trên giao diện web

Hàm mục tiêu của SVM

- Mục tiêu của SVM là tối ưu hóa khoảng cách giữa các điểm dữ liệu của hai lớp.
- Hàm mục tiêu được xây dựng sao cho margin giữa hai lớp là lớn nhất, đồng thời đảm bảo rằng không có điểm nào bị sai phân loại.
- Hàm mục tiêu của SVM là tối thiểu hóa hàm sau:

$$(w, b) = \arg\max_{w,b} \left\{ \min_n y_n \frac{(w^T x_n + b)}{\|w\|_2} \right\} = \arg\max_{w,b} \left\{ \frac{1}{\|w\|_2} \min_n y_n \left(w^T x_n + b \right) \right\}$$

Ảnh minh hoạ



Hình 1: Mô tả thuật toán SVM

Hàm mục tiêu của SVM

- Nếu dữ liệu huấn luyện có thể phân tách tuyến tính, có thể chọn hai siêu phẳng song song để phân tách hai lớp dữ liệu, sao cho khoảng cách giữa chúng là lớn nhất có thể.
- Khu vực được giới hạn bởi hai siêu phẳng này được gọi là "margin" (biên).Siêu phẳng có margin lớn nhất là siêu phẳng nằm ở giữa hai siêu phẳng này.

Hàm mục tiêu của SVM

Với một bộ dữ liệu đã chuẩn hóa hoặc chuẩn hóa, các siêu phẳng này có thể được mô tả bằng các phương trình:

- $$\mathbf{w}^T\mathbf{x} + b = 1$$

  (mọi điểm trên hoặc phía trên ranh giới này thuộc về một lớp với nhãn 1)

- $$\mathbf{w}^T\mathbf{x} + b = -1$$

  (mọi điểm trên hoặc phía dưới ranh giới này thuộc về lớp còn lại, với nhãn -1).

Về mặt hình học, khoảng cách giữa 2 siêu phẳng này là $\frac{2}{\|w\|}$ nên ta cần tối thiểu hóa $\|w\|$.

Ràng buộc hàm mục tiêu

Cũng cần ngăn không cho các điểm dữ liệu rơi vào margin, vì vậy phải thêm vào ràng buộc sau: với mỗi i, nếu $y_i = 1$, thì phải có:

$$w^T x_i + b \geq 1$$

hoặc nếu $y_i = -1$, thì phải có:

$$w^T x_i + b \leq -1$$

Ràng buộc này yêu cầu mỗi điểm dữ liệu phải nằm ở phía đúng của margin. Hay có thể viết lại là:

$$y_i(w^T x_i + b) \geq 1, \quad \forall 1 \leq i \leq n$$

Bài toán tối ưu

Tóm lại, ta cần giải bài toán tối ưu:

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

với ràng buộc:

$$y_i(w^T x_i + b) \geq 1 \quad \forall i \in \{1, \ldots, n\}$$

Cần tìm w và b

Hàm Lagrange

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i \left(y_i(w \cdot x_i + b) - 1\right)$$

$\lambda_i \geq 0$ là các hệ số Lagrange

Điều kiện KKT

$$\lambda_i \geq 0 \quad \forall i = 1, \ldots, n \tag{1}$$

$$y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) - 1 \geq 0 \quad \forall i = 1, \ldots, n \tag{2}$$

$$\lambda_i[y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) - 1] = 0 \quad \forall i = 1, \ldots, n \tag{3}$$

Phương trình Lagrange và điều kiện KKT

- $\forall i : \lambda_i = 0$ hoặc $y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) = 1$.
- Các điểm sao cho $\lambda_i > 0$ nằm trên margin được gọi là các vector hỗ trợ(support vectors)

Điều kiện KKT

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \lambda_i y^{(i)} \mathbf{x}^{(i)} = \mathbf{0} \tag{4}$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{n} \lambda_i y^{(i)} = 0 \tag{5}$$

Từ (4):

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i y^{(i)} \mathbf{x}^{(i)} \tag{6}$$

Từ (5):

$$\sum_{i=1}^{n} \lambda_i y^{(i)} = 0 \tag{7}$$

Điều kiện KKT

$$
\begin{aligned}
L(w, b, \lambda_1, \ldots, \lambda_n) &\equiv \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \lambda_i \{y^{(i)}(w^\top x^{(i)} + b) - 1\} \\
&= \frac{1}{2}\|w\|^2 - \underbrace{\sum_{i=1}^{n} \lambda_i y^{(i)} x^{(i)\top}}_{=w^\top} w - \underbrace{\sum_{i=1}^{n} \lambda_i y^{(i)}}_{=0} b + \sum_{i=1}^{n} \lambda_i
\end{aligned}
$$

Điều kiện KKT

$$\begin{aligned}
L(w, b, \lambda_1, \dots, \lambda_n) &\equiv \frac{1}{2}\|w\|^2 - \|w\|^2 + \sum_{i=1}^{n} \lambda_i \\
&= \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\|w\|^2 \\
&= \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)\top} x^{(j)} \\
&\equiv \tilde{L}(\lambda_1, \dots, \lambda_n)
\end{aligned}$$

Quy về bài toán

Có thể quy về bài toán tìm:

$$\hat{\lambda} = \arg\max_{\lambda} \tilde{L}(\lambda) \tag{8}$$

$$= \arg\max_{\lambda} \left\{ \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\,T} \mathbf{x}^{(j)} \right\} \tag{9}$$

$$\text{với điều kiện } \lambda_i \geq 0, \sum_{i=0}^{n} \lambda_i y^{(i)} = 0, (i = 1, 2, \ldots, n). \tag{10}$$

Ước lượng $\hat{\lambda}$ bằng phương pháp Gradient Descent

Nguyên lý phương pháp

- Phương pháp Gradient Descent được sử dụng để tìm nghiệm tối ưu $\hat{\lambda}$
- Các giá trị tham số ban đầu $\lambda^{[0]}$ được đặt ngẫu nhiên
- Cập nhật theo hướng gradient (vì đây là bài toán tối đa hóa):

$$\lambda^{[t+1]} = \lambda^{[t]} + \eta \tilde{L}(\lambda)\lambda$$

- η là tốc độ học (learning rate)

Biểu diễn vector cho bài toán SVM

Ma trận dữ liệu và vector

$$\mathbf{X}_{[n\times p]} = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_p^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n)} & x_2^{(n)} & \cdots & x_p^{(n)} \end{pmatrix} = \begin{pmatrix} -\mathbf{x}^{(1)\,T}- \\ -\mathbf{x}^{(2)\,T}- \\ \vdots \\ -\mathbf{x}^{(n)\,T}- \end{pmatrix}$$

$$\mathbf{y}_{[n\times 1]} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}, \quad \lambda_{[n\times 1]} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}$$

Ma trận H và phép nhân Hadamard

Định nghĩa ma trận H

$$\mathbf{H}_{[n\times n]} \equiv \mathbf{y}_{[n\times 1]}\mathbf{y}^T_{[1\times n]} \odot \mathbf{X}_{[n\times p]}\mathbf{X}^T_{[p\times n]}$$

- $\odot$ là tích Hadamard (phép nhân từng phần tử)
- Các phần tử của ma trận: $(H)_{ij} = y^{(i)}y^{(j)}\mathbf{x}^{(i)\,T}\mathbf{x}^{(j)}$
- H là ma trận đối xứng

Viết lại hàm Lagrangian

Biểu diễn hàm Lagrangian dưới dạng vector

$$\begin{aligned}\tilde{L}(\lambda) &\equiv \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\,T} \mathbf{x}^{(j)} \\ &= \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j (H)_{ij} \\ &= \|\lambda\| - \frac{1}{2} \lambda^T H \lambda\end{aligned}$$

Với $\|\lambda\| = \sum_{i=1}^{n} \lambda_i$ là tổng các phần tử của vector λ

Tính toán vector gradient

Đạo hàm của hàm Lagrangian theo λ

$$\tilde{L}(\lambda)\lambda = \lambda\|\lambda\| - \frac{1}{2}\lambda\lambda^T H\lambda$$
$$= \mathbf{1} - H\lambda$$

Trong đó $\mathbf{1}$ là vector cột với tất cả các thành phần bằng 1: $\mathbf{1} = (1, 1, \ldots, 1)^T$

Quy tắc cập nhật trong Gradient Descent

Quy tắc cập nhật cho nhân tử λ

$$\lambda^{[t+1]} = \lambda^{[t]} + \eta(\mathbf{1} - H\lambda^{[t]})$$

- Cập nhật này được lặp đi lặp lại cho đến khi hội tụ
- Cần đảm bảo ràng buộc $\lambda_i \geq 0$ bằng cách cắt giá trị âm về 0

Tìm $\hat{\mathbf{w}}$ từ $\hat{\lambda}$

Tính vector trọng số $\hat{\mathbf{w}}$

Từ điều kiện KKT, ta có:

$$\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\lambda}_i y^{(i)} \mathbf{x}^{(i)}$$

- Từ điều kiện KKT (3):

$$\hat{\lambda}_i = 0, \text{ hoặc } y^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)} + b) - 1 = 0$$

- Dữ liệu $\mathbf{x}^{(i)}$ được phân loại:

$$\begin{cases} \hat{\lambda}_i \neq 0 \Leftrightarrow \mathbf{x}^{(i)} \text{ là vector hỗ trợ,} \\ \hat{\lambda}_i = 0 \Leftrightarrow \mathbf{x}^{(i)} \text{ không phải là vector hỗ trợ.} \end{cases}$$

Tìm $\hat{\mathbf{w}}$ từ $\hat{\lambda}$

Tính vector trọng số $\hat{\mathbf{w}}$

Từ điều kiện KKT, ta có:

$$\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\lambda}_i y^{(i)} \mathbf{x}^{(i)}$$

- Chỉ tính tổng trên các vector hỗ trợ:

$$\hat{\mathbf{w}} = \sum_{\mathbf{x}^{(i)} \in S} \hat{\lambda}_i y^{(i)} \mathbf{x}^{(i)}$$

Tìm $\hat{b}$ từ các vector hỗ trợ

Tính tham số $\hat{b}$

$$\begin{aligned}\hat{b} &= \frac{1}{y^{(i)}} - \hat{\mathbf{w}}^T \mathbf{x}^{(i)} \\ &= y^{(i)} - \hat{\mathbf{w}}^T \mathbf{x}^{(i)} \quad (\text{vì } y^{(i)} = 1 \text{ hoặc } -1)\end{aligned}$$

- Trong thực tế, để giảm sai số, tính trung bình trên tất cả các vector hỗ trợ:

$$\hat{b} = \frac{1}{|S|} \sum_{\mathbf{x}^{(i)} \in S} \left(y^{(i)} - \hat{\mathbf{w}}^T \mathbf{x}^{(i)} \right)$$

- $|S|$ là số lượng vector hỗ trợ

Tóm tắt thuật toán Gradient Descent cho SVM

1. **Khởi tạo:** $\lambda^{[0]}$ ngẫu nhiên, tính ma trận H
2. **Lặp:** Cập nhật λ theo công thức:

$$\lambda^{[t+1]} = \lambda^{[t]} + \eta(\mathbf{1} - H\lambda^{[t]})$$

3. **Áp dụng ràng buộc:** $\lambda_i \geq 0$
4. **Tìm vector hỗ trợ:** $S = \{i : \lambda_i > 0\}$
5. **Tính tham số mô hình:**

$$\hat{\mathbf{w}} = \sum_{\mathbf{x}^{(i)} \in S} \hat{\lambda}_i y^{(i)} \mathbf{x}^{(i)}$$

$$\hat{b} = \frac{1}{|S|} \sum_{\mathbf{x}^{(i)} \in S} (y^{(i)} - \hat{\mathbf{w}}^T \mathbf{x}^{(i)})$$

Mô tả bằng mã giả

Algorithm 1 The Pegasos algorithm.

Inputs: a list of example feature vectors $\boldsymbol{X}$
a list of outputs Y
regularization parameter λ
the number of steps T

$\boldsymbol{w} = (0, \ldots, 0)$
for t in $[1, \ldots, T]$
  select a position i randomly
  $\eta = \frac{1}{\lambda \cdot t}$
  score = $y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i)$
  if score < 1
    $\boldsymbol{w} = (1 - \eta \cdot \lambda) \cdot \boldsymbol{w} + (\eta \cdot y_i) \cdot \boldsymbol{x}_i$
  else
    $\boldsymbol{w} = (1 - \eta \cdot \lambda) \cdot \boldsymbol{w}$
the end result is $\boldsymbol{w}$

Hình 2: Mô tả thuật toán Pegasos cơ bản bằng mã giả

Bước Chiếu

Giới hạn tập hợp các nghiệm khả thi trong phạm vi $\frac{1}{\sqrt{\lambda}}$. Để thực hiện điều này, cập nhật $\mathbf{w}_t$ sau mỗi vòng lặp:

$$\mathbf{w}_{t+1} \leftarrow \min\left(1, \frac{1}{\sqrt{\lambda}\|\mathbf{w}_{t+1}\|}\right)\mathbf{w}_{t+1}.$$

Hàm mục tiêu

Cần tìm vector trọng số **w** để tối thiểu hóa hàm mục tiêu sau:

$$f(\mathbf{w}, \mathbf{X}, \mathbf{Y}) = \frac{\lambda}{2} \cdot \|\mathbf{w}\|^2 + \frac{1}{|\mathbf{Y}|} \cdot \sum_i \mathrm{Loss}(\mathbf{w}, \mathbf{x}_i, y_i)$$

Đối với thuật toán SVM, hàm mất mát là hàm mất mát hinge:

$$\mathrm{Loss}(\mathbf{w}, \mathbf{x}_i, y_i) = \max(0, 1 - y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i))$$

Hàm mất mát Hinge

Hàm mất mát hinge có thể được viết rõ ràng hơn như sau:

$$\text{Loss}(\mathbf{w}, \mathbf{x}_i, y_i) = \begin{cases} 1 - y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i) & \text{nếu } y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i) < 1 \\ 0 & \text{khác} \end{cases}$$

Điều mà Pegasos thực hiện là áp dụng một thuật toán tối ưu hóa để tìm **w** tối thiểu hóa hàm mục tiêu f:

$$f(\mathbf{w}; A_t) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{k}\sum_{i \in A_t} \ell(\mathbf{w}; (\mathbf{x}_i, y_i)).$$

với k là số lượng tập mini-batch.

1. Giới thiệu
2. Lý thuyết
3. Cài đặt
    - Tiền xử lý dữ liệu
4. Results

Import thư viện

Figures

Hình 3: Logo of the university.

- different themes
- different themes
- different themes
- different themes

Cảm ơn cô đã lắng nghe!